Chapter 1

MULTI-TARGET ASSIGNMENT AND PATH PLANNING FOR GROUPS OF UAVS *

Theju Maddula
Ali A. Minai
Marios M. Polycarpou
Department of Electrical & Computer Engineering and Computer Science
University of Cincinnati
Cincinnati, OH 45221

Abstract Uninhabited autonomous vehicles (UAVs) have many useful military applications, including reconnaissance, search-and-destroy, and search-and-rescue missions in hazardous environments such as battle£elds or disaster areas. Recently, there has been considerable interest in the possibility of using large teams (swarms) of UAVs functioning cooperatively to accomplish a large number of tasks (e.g., £nding and attacking targets). However, this requires the assignment of multiple spatially distributed tasks to each UAV along with a feasible path that minimizes effort and avoids threats.

In this work, we consider an extended environment with M UAVs, N targets and P threats. The goal is to assign all the targets to the UAVs so as to minimize the maximum path length, divide work equitably among the UAVs, and limit the threat faced by each UAV. We use a four stage approach to address this problem. First, a Voronoi tessellation around the threats is used to create a graph of potential paths and waypoints. The segments of this graph are then systematically removed by a threat/cost-based thresholding process to obtain a feasible set of path elements. In the second stage, this reduced graph is searched to identify short paths between tasks and from UAVs to tasks. In the third stage, initial paths for UAVs are constructed using a semi-greedy heuristic that divides tasks equally among UAVs. Finally, in the fourth stage, this initial assignment is re£ned using spatially constrained exchange of sub-paths among UAVs. A direct method for obtaining paths of approximately equal length is also considered.

*This work was supported by the AFRL/VA and AFOSR Collaborative Center of Control Science (Grant F33615-01-2-3154).

1. Introduction

Unmanned vehicles — airborne, undersea, and land-based — have become an integral part of the battle£eld environment. They also have civilian applications such as disaster relief, environmental monitoring and planetary exploration. There has recently been considerable interest in making these unmanned vehicles completely autonomous, giving rise to the research area of *unmanned autonomous vehicles* or *UAV*s. These are usually seen as rather simple vehicles, acting cooperatively in teams to accomplish dif£cult missions in dynamic, poorly known or hazardous environments (Passino, 2002; Chandler and Pachter, 1998; Chandler et al., 2000; Chandler and Pachter, 2001; Chandler et al., 2001; Chandler, 2002; Beard et al., 2000; Beard et al., ; Bellingham et al., 2001; Jacques, 1998; Li et al., 2002; McLain and Beard, 2000; McLain et al., 2002; Moitra et al., 2001; Polycarpou et al., 2001; Polycarpou et al., 2002; Schumacher et al., 2001).

Typically, the mission to be accomplished by a group of UAVs involves completing a set of tasks spread over an extended region. The UAVs must reach each task location — possibly under temporal order constraints — and accomplish it while avoiding a spatially distributed set of threats or obstacles. In a military search-and-destroy mission, the tasks may correspond to targets to be attacked, and the threats may be enemy radar or anti-aircraft batteries. One of the primary challenges, then, is to assign the UAVs to the known tasks/targets and to plan paths for all UAVs such that the overall mission completion time is minimized and the UAVs are exposed to as little threat as possible. This is, in fact, a very complex optimization problem known to be NP-complete. It is very similar to the well-known vehicle routing problem (VRP), for which several heuristic methods have been presented (Schouwenaars et al., 2001; Tan et al., 2002). In practice, the UAV problem can be somewhat more complex if multiple task and UAV types exist, with different capabilities for each UAV type. In this paper, we report on some methods to address the simpler version where there is only one type of task and all UAVs are identical.

The approach we follow is motivated by the seminal work of Chandler and Pachter, and their collaborators (Chandler and Pachter, 1998; Chandler et al., 2000; Chandler and Pachter, 2001; Chandler et al., 2001; Chandler, 2002; Schumacher et al., 2001), and is also related closely to recent work by several other researchers (Beard et al., 2000; Beard et al., ; Bellingham et al., 2001; Jacques, 1998; Li et al., 2002; McLain and Beard, 2000; McLain et al., 2002). A comprehensive overview of the research problems associated with UAV teams is available in (Passino, 2002).

1.1. Scenario

We begin with a team of UAVs $U_i \in \mathbf{U}$, $i = 1, ..., N_U$, in a 2-dimensional environment. The position of UAV U_i at time t is given by $(x_i^U(t), y_i^U(t))$. All UAVs are modelled as point objects that move with constant speed and without restriction on turning and maneuvering. The environment has targets, $T_i \in \mathbf{T}$, $i = 1, ..., N_T$, and threats $D_i \in \mathbf{D}$, $i = 1, ..., N_D$ distributed across it. The position of target T_i is denoted by (x_i^T, y_i^T), while that of threat D_i is given by (x_i^D, y_i^D). Both targets and threats are assumed to be stationary. We assume that all targets and threats are known *a priori*. All threats are assumed to be equally lethal.

Each UAV, U_i, is to be assigned a path, $\{P_i\}$, given by

$$P_i = \{(x_i^U(0), y_i^U(0)), (x_{i_1}^T, y_{i_1}^T), ..., (x_{i_{n_i}}^T, y_{i_{n_i}}^T)\} \quad (1)$$

indicating the targets it is to visit in sequence starting from its initial position. The set $\Theta_i = \{(T_{i_1}, T_{i_2}, ..., T_{i_{n_i}}\}$ is termed the *target set* for U_i. The sub-path between the kth and $k+1$th target locations in P_i is termed *leg* L_i^k, while the sub-path from $(x_i^U(0), y_i^U(0))$ to $(x_{i_1}^T, y_{i_1}^T)$ is called leg L_i^0. Thus, path P_i can be represented also as $P_i = \{L_i^0, L_i^1, ..., L_i^{n_i}\}$. Each leg has two associated parameters: 1) A *leg length*, λ_i^k, giving the distance to be travelled on that leg; and 2) A *leg risk*, ρ_i^k, which is a normalized measure of threat along that leg. This allows us to de£ne $\lambda_i = \sum_{k=0}^{n_i} \lambda_i^k$ as the *total length* of P_i and $\rho_i = \max_{k=0}^{n_i} \rho_i^k$ as its *maximal risk*.

The *target assignment and path planning problem* is speci£ed as follows:

Find target sets for all UAVs such that:

1 Each target is assigned to some UAV, i.e., $\bigcup_{i=1}^{N_U} \Theta_i = \mathbf{T}$.

2 No target is assigned to multiple UAVs, i.e., $\Theta_i \bigcap \Theta_j = \Phi \; \forall i = j$, $i, j \in \{1, ..., N_U\}$.

3 The total path length for all UAVs is minimized, i.e., $\max_{i=1}^{N_U} \lambda_i$ is minimized.

4 UAV loads are balanced, i.e., $|\max_{i=1}^{N_U} \lambda_i - \max_{i=1}^{N_U} \lambda_i|$ is minimized.

5 $\max_{i=1}^{N_U} \rho_i < \theta_r$, where θ_r is a risk tolerance threshold parameter.

Here, we are assuming implicitly that time is needed only to travel between targets, and not for performing tasks at target locations. However, this can be addressed by adding a *task performance time* to each leg.

This formulation seeks to minimize the completion time for the mission (visiting all known targets) while keeping risk below threshold and balancing

UAV load. Other objectives could be used, e.g., minimizing mean λ_i. Other constraints could be added, e.g., maintaining a minimum UAV separation or uniformity of coverage.

The problem de£ned here is clearly very hard. It is essentially equivalent to a multiple vehicle routing problem. We address it by dividing the problem into two phases:

- In the £rst phase, we obtain a *feasible paths graph (FPG)* that gives sub-paths that are "feasible" or "promising" as legs of potential paths. In particular, we eliminate all legs that would have risk greater than θ_r, thus satisfying one of the objectives by construction. This is essentially the "satis£cing" approach used in (McLain and Beard, 2000; McLain et al., 2002) More broadly, the goal is to make this graph as sparse as possible without excluding good sub-paths or making some targets unreachable. This greatly reduces the search space for the overall optimization problem.

- During the second phase, we construct paths based on the graph obtained during Phase I using just the path length as the cost. Since all UAVs move with equal speed, optimizing over path length can be used to minimize total completion time and balance UAV loads.

1.2. Constructing the FPG

The FPG can be constructed as follows:

1 Obtain a Voronoi tessellation of the environment based on all known threats. The edges, e_{pq}, of the Voronoi tessellation form a graph, $\bar{G}_W$, with each intersection point of edges comprising a node, v_q. We term this the *waypoint graph (WG)*, since each node in it is a potential waypoint, and each edge is the minimal risk subpath between a pair of threats.

2 From each target, add a line to the m nearest waypoints. Add these lines as edges to the waypoint graph to obtain the *augmented waypoint graph, (AWG)*, $G_W \equiv \{V_G, E_G)\}$. The targets thus become nodes in the AWG, but are called *target nodes* to distinguish them from the waypoint nodes.

3 To each edge, e_{pq}, in G_W, assign a risk, r_{pq} based on the distance of various points on the edge from nearby threats (Chandler et al., 2000; Beard et al., 2000; Beard et al., ; McLain and Beard, 2000):

$$r_{pq} = l_{pq} \sum_{j=1}^{N_D} \{ \frac{1}{d^4_{1/6,pq,j}} + \frac{1}{d^4_{1/2,pq,j}} + \frac{1}{d^4_{5/6,pq,j}} \} \qquad (2)$$

where l_{pq} is the length of the edge, $d_{f,pq,j}$ is the distance between threat D_j and the point on e_pq that is a fraction f of the total edge length from point p. This is an approximation for the more accurate risk cost that would be obtained by integrating along the entire edge.

4 Delete all edges in the AWG that have risk higher than a threshold, θ_r. This gives the *reduced edge graph (REG)*.

5 Using constrained search on the REG, obtain the K shortest paths from each UAV, U_i, to all targets within distance θ_d, and between all pairs of targets that are "suf£ciently close". Calculate the length, λ_P, and risk, ρ_P, of each path, P:

$$\lambda_P = \sum_{e_{pq} \in P} l_{pq} \tag{3}$$

$$\rho_P = \max_{e_{pq} \in P} r_{pq} \tag{4}$$

6 Construct a graph, $F \equiv \{V_F, E_F\}$, with two types of nodes: targets and UAVs. Thus, $V_F \subseteq V_G$. If there is a path, P, between nodes p and q in the REG with $\lambda_P \leq \theta_p$, there is an edge, $\epsilon_{pq} \in E_F$, between them in F, labelled by the cost of the path. F is the FPG, and is the basis of path planning.

The goal of the entire process outlined above is to obtain a graph that includes only feasible and "good" paths from UAVs to targets and between targets, so that the search for multi-target paths focuses only on these — eliminating the vast majority of paths that make the problem hard. However, the use of thresholds at various steps requires care lest the graph become disconnected or leave too few options. We, therefore, iterate over the algorithm above, adjusting the thresholds heuristically until a reasonably dense FPG with at least ϕ edges to each target is obtained.

1.3. Assignment of Multiple-Target Paths

For the second stage of the assignment process, we consider three methods. The £rst is a naive greedy method while the other two are a little more sophisticated.

Method 1: Target Equalization (TE). In this method, the objective is to make sure that each UAV gets the same number of targets. It is done as follows:

I Until all targets are assigned, repeat:

Ia Assign each UAV to its best available target.

Ib Resolve con¤icts in favor of the UAV with lower cost.

Ic Iterate over UAVs and targets until each UAV is assigned a target in the current cycle, or no targets are left.

Clearly, this is a greedy method, and is not likely to produce a good solution. However, it does provide a baseline for evaluating other methods.

Path Equalization (PE). The use of Target Equalization causes some UAVs to have short paths while others end up with very long paths, thus increasing the mission completion time. The Path Equalization method tries to make the paths of each UAV nearly the same length. It proceeds as follows:

I Assign a £rst target to each UAV the same way as in Target Equalization.

II Until all targets are assigned, repeat:

IIa Choose the UAV with the shortest cumulative path so far.

IIb Assign the closest available target to the chosen UAV.

IIc Update the cumulative path length for the chosen UAV.

This method differs from TE mainly in that it chooses the next UAV for assignment based on current loading rather than in a £xed order through all UAVs. This is an explicit attempt to balance the loading. However, this also, indirectly, leads to the reduction of the longest path to keep it close to the average over all UAVs.

Two Stage Method. When using the Path Equalization method, the options available to a particular UAV decrease with the number of available targets. This is not a problem while the number of available targets is still large, but towards the later stages of the assignment process, it can lead to a situation where the UAV with the shortest cumulative path at the time of selecting the next target has much poorer options than another UAV with a somewhat longer cumulative path. This can lead to a drastic and irreversible imbalance in path lengths — the very problem that PE was designed to correct.

We address this issue by using a two-stage assignment process controlled by a time-varying parameter, $\xi(t)$, de£ned as:

$$\xi(t) \equiv \frac{N_{available}(t)}{N_U} \tag{5}$$

where $N_{available}(t)$ is the number of targets still unassigned at step t.

Stage I: When $\xi(t) \geq \theta_\xi$, any selected UAV is likely to have many options for its next assignment, and the PE algorithm is followed. Here, θ_ξ is a threshold.

Stage II: When $\xi(t) < \theta\xi$, many UAVs are likely to have very limited choices, and the simple PE approach could lead to poor assignments. In this case, the assignment is done as follows:

I Set Ψ as the set of all UAVs.
Set $\lambda^* = Z$, where Z is a very large number.
Set $i^* = 1$.

II Until an assignment is accepted or Ψ is empty, repeat:

IIa Pick the $U_i \in \Psi$ with the shortest cumulative path length, $\lambda_i(t)$.

IIb Determine the nearest available target for U_i, and calculate the resulting updated path length, $\lambda_i^+(t)$. If $\lambda_i^+ < \lambda^*$, set $i^* = i$.

- If $\lambda_i^+(t) < \max_{j\in\Psi} \lambda_j(t)$, i.e., the resulting updated path would not exceed the longest current path, accept the assignment.
- else, remove U_i from Ψ.

III If no assignment was made in step II, choose the assignment for U_{i^*}.

Heuristically, the argument underlying the procedure is as follows. At the beginning of the assignment, when each UAV has many target options, the focus is on limiting the search by an intelligent choice of which UAV to update at each step (PE). However, towards the later stages of the process, the choices for each UAV are limited, and it is both better and computationally more feasible to consider several — even all — UAVs for each update rather than myopically picking the one with the shortest path so far. If there is no good choice, the least bad choice is made (step III).

Refinement. The three algorithms described above produce reasonable solutions, but these can be improved further. We do so through a spatially constrained partial path exchange procedure described below. For the description, we need to de£ne a concept called a *potential point of exchange (PPE)* as follows. Given path P_i currently assigned to U_i and P_j assigned to U_j: If vertices $v_1 \in P_i$ and $v_2 \in P_i$ are such that $||v_1 - v_2|| < \theta_e$, we term $[v_1, v_2]$ a potential point of exchange. Based on this, we de£ne four *exchange operators*:

- **Operator 1:** The subpaths of P_i and P_j starting at v_i and v_j, respectively, are exchanged.

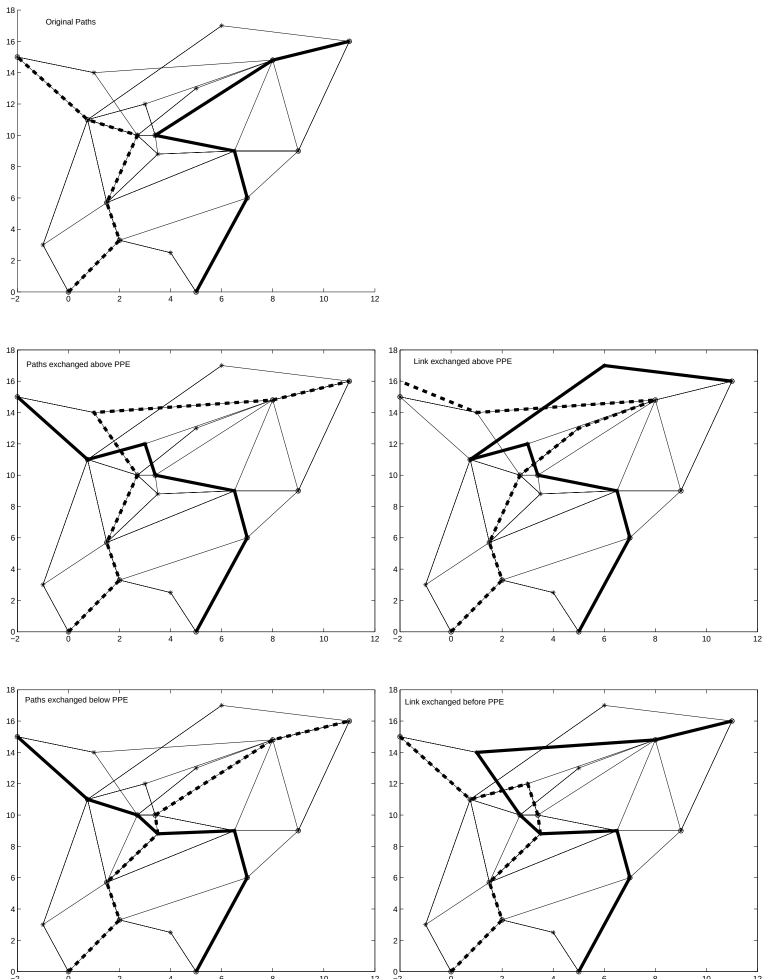

Figure 1.1. The £rst graph shows two assigned paths on an FPG. The next four graphs show the effect of applying each of the exchange operators to the original paths. Note that not all operators produce better options. The ones that are worse than the original will be rejected

- **Operator 2:** The subpaths of P_i and P_j, including the segments terminating at v_i and v_j, respectively, are exchanged.

- **Operator 3:** The segments of P_i and P_j starting at v_i and v_j, respectively, are exchanged.

- **Operator 4:** The segments of P_i and P_j ending at v_i and v_j, respectively, are exchanged.

The exchange operators are illustrated in Figure 1. Operators 1 and 2 exchange whole subpaths, and are the major drivers of search. Operators 3 and 4 serve mainly to remove “kinks” in existing pairs of paths — cases where the paths cross and then cross again within a short distance. It should be noted that exchanging paths typically requires a local search on the FPG because the graph is not fully connected. If the exchange being considered is not possible because of this, it is rejected automatically.

The procedure begins with the assignment produced by one of the above algorithms, and iterates over the following procedure until a stopping criterion is met:

I Find two *candidate paths*, P_i and P_j that have a PPE.

II Apply one of the exchange operators picked at random to the candidate paths.

III If the new paths are more equal than the old paths, keep the new ones. Else, do not accept the exchange.

Typical stopping criteria are a suf£cient frequency of rejection in step III or a limit on number of iterations.

2. Simulation Results

We evaluated the performance of each assignment procedure with and without re£nement using a size 40×40 continuous 2-dimensional environment with 40 randomly placed threats and 8 UAVs. The number of targets is varied from 8 to 48 in increments of 8. The UAVs start from randomly chosen locations. All data is averaged over 20 independent simulations with new target, threat and UAV locations.

Figure 2 shows the maximum path length as a function of the number of targets. As expected, TE has the poorest performance while the 2-stage method does the best. Also, re£nement helps quite substantially in both cases.

Figure 3 shows the improvement produced by re£nement with each of the three basic procedure. The most substantial improvement is in TE, while the improvement for PE and the two stage method are similar. In fact, applying re£nement to TE makes it comparable to — possibly even better than — PE with re£nement.

Finally, Figure 4 shows the difference between the average path length and the maximum path length. Again, it is clear that the two stage method with re£nement does the best. It is also worth noting that, for PE and the two stage method, this quantity declines with increasing number of targets. However, it levels off for PE while continuing to decline for the 2 stage method, indicating a clear advantage for the latter.

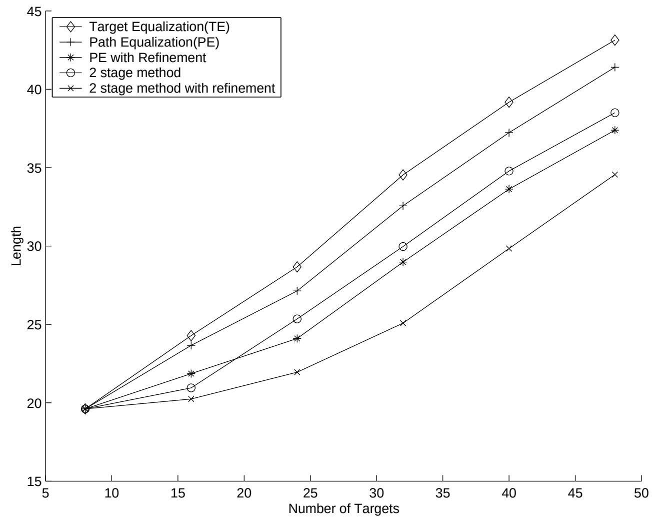

Figure 1.2. Maximum path length, averaged over 20 independent simulations. The environment is a size 40×40 continuous 2-dimensional space with 8 UAVs and 40 threats. Distances are Euclidean.

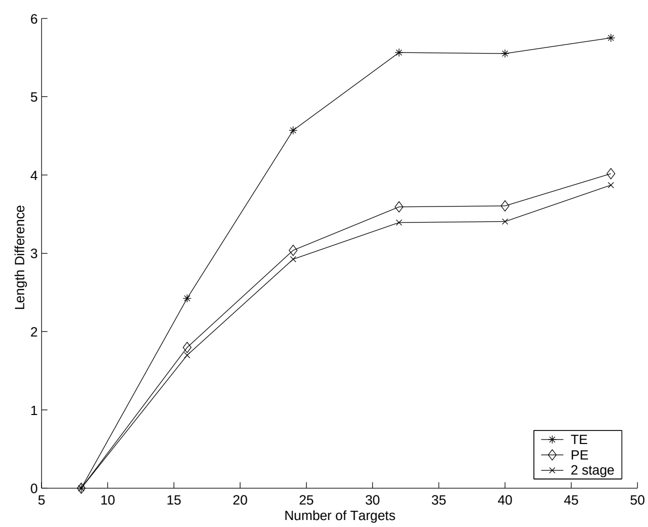

Figure 1.3. Improvement in the maximum path length due to re£nement.

3. Conclusion and Future Work

In this paper, we have compared several heuristically derived methods for assigning a group of UAVs to multiple targets and planning safe paths for them. The results are encouraging, and indicate that the two stage approach, in par-

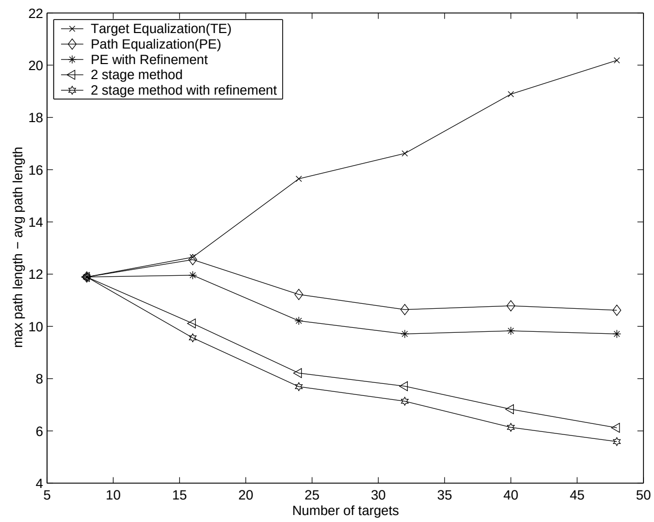

Figure 1.4. Difference between maximum and average path length.

ticular, may be a viable one. However, there is much scope for further research — especially a systematic evaluation of the parameters used. Issues currently under investigation are: 1) More sophisticated re£nement processes such as genetic recombination and selection or simulated annealing; 2) Improvements to the two stage approach, studying the effect of varying $\xi(t)$ on the solution, and considering other parameters; and 3) Comparing the results with those obtained using methods used by other researchers.

References

Beard, R., McLain, T., and Goodrich, M. (2000). Coordinated target assignment and intercept for unmanned air vehicles. *Proc. ICRA'2000*, pages 2581–2586.

Beard, R., McLain, T., Goodrich, M., and Anderson, E. Coordinated target assignment and intercept for unmanned air vehicles. *IEEE Trans. On Robotics and Automation*.

Bellingham, J., Tillerson, M., Richards, A., and How, J. (2001). Multi-task allocation and path planning for cooperative uavs. *Conference on Coordination, Control and Optimization*.

Chandler, P. and Pachter, M. (1998). Research issues in autonomous control of tactical uavs. *Proc. ACC'1998*, pages 394–398.

Chandler, P. and Pachter, M. (2001). Hierarchical control for autonomous teams. *Proc. GNC'2001*, pages 632–642.

Chandler, P., Pachter, M., and Rasmussen, S. (2001). Uav cooperative control. *Proc. ACC'2001*.

Chandler, P., Rasmussen, S., and Pachter, M. (2000). Uav cooperative path planning. *Proc. GNC'2000*, pages 1255–1265.

Chandler, P. e. a. (2002). Complexity in uav cooperative control. *Proc ACC'2002*.

Jacques, D. (1998). Search, classi£cation and attack decisions for cooperative wide area search munitions. *Proc. Cooperative Optimization and Control Workshop*.

Li, S.-M., Boskovic, J., Seereeeram, S., Prasanth, R., Amin, R., Mehra, R., and Beard, R. a. M. T. (2002). Autonomous hierarchical control of multiple unmanned combat air vehicles (ucavs). *Proc. ACC'2002*, pages 274–279.

McLain, T. and Beard, R. (2000). Trajectory planning for coordinated rendezvous of unmanned air vehicles. *Proc. GNC'2000*, pages 1247–1254.

McLain, T., Beard, R., and Kelsey, J. (2002). Experimental demonstration of multiple robot cooperative target intercept. *Proc GNC'2002*.

Moitra, A., Szczerba, R., Didomizio, V., Hoebel, L., Mattheyses, R., and Yamrom, B. (2001). A novel approach for the coordination of multi-vehicle teams. *Proc. GNC'2001*, pages 608–618.

Passino, K. (2002). An introduction to research challenges in cooperative control for uninhabited autonomous vehicles.

Polycarpou, M., Yang, Y., and Passino, K. (2001). A cooperative search framework for distributed agents. *Proc. 2001 IEEE ISIC*, pages 1–6.

Polycarpou, M., Yang, Y., and Passino, K. (2002). Cooperative control of distributed multi-agent systems. *IEEE Control Systems Magazine*.

Schouwenaars, T., De Moor, B., Feron, E., and How, J. (2001). Mixed integer programming for multi-vehicle path planning. *Proc. ACC'2001*.

Schumacher, C., Chandler, P., and Rasmussen, S. (2001). Task allocation for wide area search munitions via network ¤ow optimization. *Proc. GNC'2001*, pages 619–626.

Tan, K., Lee, L., Zhu, Q., and Ou, K. (2002). Heuristic methods for vehicle routing problem with time windows. *Intelligent in Engineering*, pages 281–295.